Connectivitat i Percolació

Projecte d'Algorísmia • FIB-UPC • QT 2024-2025

Subgrup: 14
Pablo Calomardo Sangüesa
Alejandro Lorenzo Navarro
Andres Lucian Laptes Costan
Enric Segarra Calbet

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Índex

1. Estructura de Dades

Hem decidit representar els grafs utilitzant **llistes d'adjacències** en lloc d'una matriu d'adjacència donat que ens basem tant en criteris d'eficiència computacional i de memòria.

Estructura del graf i eficiència en l'ús d'una llista d'adjacències

1. **Cost espacial**: L'ús d'una matriu d'adjacència requeriria $O(n^2)$ espai, on n és el nombre de nodes, ja que caldria emmagatzemar la relació de cada node amb tots els altres, independentment de si estan connectats. En canvi, una **llista d'adjacències** només emmagatzema les connexions existents, requerint $O(n + m)$ espai, on m és el nombre d'arestes. Bàsicament, la matriu ocupa més espai y la llista és més eficient en termes de memòria, especialment quan el graf té una densitat baixa d'arestes respecte al nombre de nodes.
2. **Eficiència algorísmica**: Els algorismes utilitzats per a la percolació i la cerca de components connexes es tornen més simples amb una representació per llistes d'adjacència. En una matriu d'adjacència, les operacions de cerca requereixen revisar totes les parelles de nodes, amb una complexitat de $O(n^2)$. En canvi, amb una llista d'adjacències, només es recorren les connexions existents, reduint la complexitat a $O(n + m)$. Això és especialment rellevant en experiments de percolació on s'apliquen diverses simulacions, ja que redueix significativament el temps de processament.

Conclusions

Aquesta representació amb **llistes d'adjacència** permet una gestió més eficient de la memòria i del temps d'execució dels algorismes en grafs per als nostres experiments, facilitant la simulació de grans mostres necessàries per estudiar les transicions de fase. On hem decidit utilitzar com a estructura una list<pair<int, list>, on bàsicament el primer element de la parella és el vèrtex actual i la llista del segon element de la parella són les seves arestes. Vam decidir utilitzar llistes abans que vectors donat que en C++ per eliminar elements d'un vector aquest fa una còpia del vector original sense l'element que volem eliminar, a més a més, evitem problemes de memòria a l'hora de fer *push_backs*.

2. Algoritmes de percolació i de components connexes

Algoritmes de percolació

Per als algoritmes de percolació, tant d'arestes com de nodes, hem utilitzat un algoritme exactament igual, amb l'única diferència que un esborra arestes i l'altre nodes. El funcionament és el següent:

- El subprograma rep com a paràmetres el graf a percolar i la probabilitat p de percolació.
- Per a cada node/aresta, es genera un nombre real aleatori entre 0 i 1. Si aquest nombre és major que 1-p, s'elimina el/la node/aresta. Per tant, realitza una cerca de cost O(n+m).

A l'hora d'eliminar nodes o arestes el cost varia. Per a les arestes el cost és O(n+m), ja que només s'ha de realitzar una única cerca perquè ja sabem a quins nodes pertany l'aresta. Eliminar un element de la llista d'arestes té cost O(m) i, com s'han d'eliminar dues arestes, el cost de percolació d'arestes té cost $O((n+m)^2+2*m)$, per tant, el cost és $O((n+m)^2)$.

En canvi, per a eliminar nodes hem de realitzar dues cerques, una per trobar el node a eliminar i una altra per eliminar totes les arestes dels nodes que en tenen una amb el node a eliminar, com a resultat d'aquest fet el cost de cerca és de O((n+m)2). Com que el cost d'eliminar d'una llista d'arestes és O(m) i hem d'eliminar m arestes, el cost de l'eliminació de totes les arestes del node a eliminar és O(m2). Per una altra banda, el cost d'eliminar de la llista de nodes és O(n), llavors el cost total de l'algoritme de percolació de nodes és $O((n+m)^3+m^2+n)$, per consegüent, el cost és $O((n+m)^3)$.

Algoritme per a comptar les components connexes d'un graf

Per a l'algorisme que compta components connexes d'un graf no dirigit hem decidit utilitzar una estratègia amb DFS (cerca en profunditat). El funcionament és el següent:

- Crea un vector de booleans amb la mida dels nodes, inicialitzat tot a *false* (Cost O(n)).
- Aprofitant que els nodes del graf s'identifiquen numèricament començant pel zero, creixent d'un en un, s'inicialitza una variable que comptabilitzarà el nombre de DFS que es realitzen fins que tots els nodes del graf han sigut visitats, és a dir, per cada node que visiti el DFS es marcarà com a visitat al vector de booleans anterior. El nombre de DFS que es realitzen equival al nombre de components connexes del graf.

Per tant, el cost serà el d'una cerca, ja que només recorrerà tots els vectors i les arestes un cop. Llavors el cost de l'algoritme és O(n+(n+m)), que equival a un cost O(n+m).

3. Experimentació amb graelles quadrades

Descripció de l'experiment

En aquest apartat se'ns demana dissenyar un experiment per estudiar la probabilitat de transició de fase amb relació a la connectivitat del graf per a un tipus específic de grafs, les graelles quadrades. Per a fer-ho, hem decidit experimentar amb graelles de diferents mides, en un rang de 2*2 nodes fins a 15*15 nodes, per tant, un màxim de 225 nodes.

Per a les graelles quadrades hem considerat dur a terme dos experiments, cadascun dos cops, un per a la percolació de nodes i un per a la percolació d'arestes. El primer experiment consisteix a realitzar, per a 200 grafs de mida aleatòria (dins del rang descrit anteriorment), la percolació pertinent amb un rang de probabilitats tal que: [0, 0.005, 0.01, 0.015, … , 0.985, 0.99, 0.995, 1.00] i veure si té una o més components connexes. Això ho realitzarem 15 cops i guardarem la mitja dels 15 experiments per a visualitzar un gràfic amb els resultats, podent comprovar així l'existència de transició de fase o no.

En el cas que la transició de fase existeix volem observar com varia segons la mida del graf. Per aquest motiu, el segon experiment consistirà en, un cop per cada mida (de 2*2 fins a 15*15), percolar amb les probabilitats anteriors un únic de graf 200 cops, fer la mitja dels resultats i crear el gràfic, igual que en l'anterior. Així podrem visualitzar la diferència, si existeix, de la transició de fase segons la mida de la graella quadrada.

A més a més, hem dut a terme un petit experiment addicional sobre les components annexes que es pot consultar a l'annex.

Per aquest dos experiments hem hagut d'implementar:

1. Un Generador de Grafs graella quadrats
2. Un controlador que ens permet escollir quin experiment realitzar, el qual crida les percolacions necessàries i guarda les dades en un .csv.
3. Un compactador d'estadístiques, és a dir, un codi que fa la mitja de les dades dels .csv i ho guarda a un .csv definitiu.
4. Un visualitzador de fitxers CSV.

També hem fet un visualitzador de grafs per a poder entendre si el programa està percolant i comprovant bé els grafs.

Experiment 1:

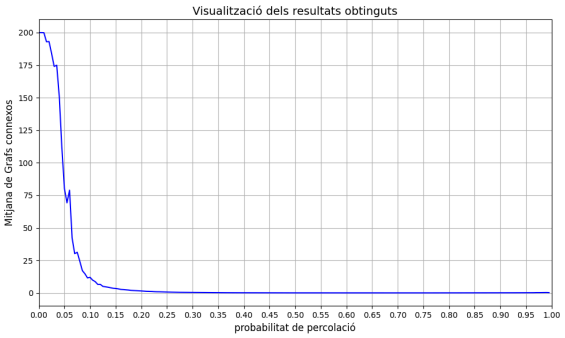
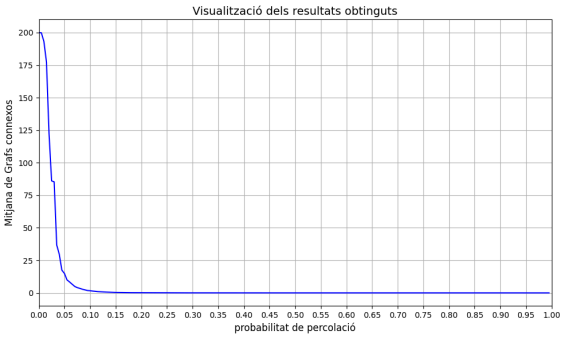
El funcionament de l'experiment és el següent:

1. Generem un conjunt aleatori de 200 grafs graella quadrats de mida entre 2*2 i 15*15
2. Apliquem les percolacions de nodes a cada un dels grafs per cada probabilitat dins del rang mencionat anteriorment i observem per a cada percolació si el graf només conté una component connexa o en té més d'una.
3. Per a cada probabilitat, operem aquesta fórmula:

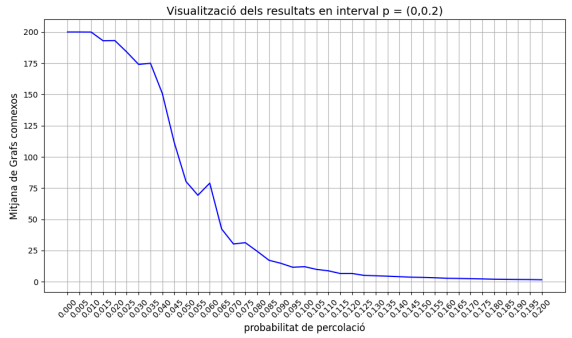
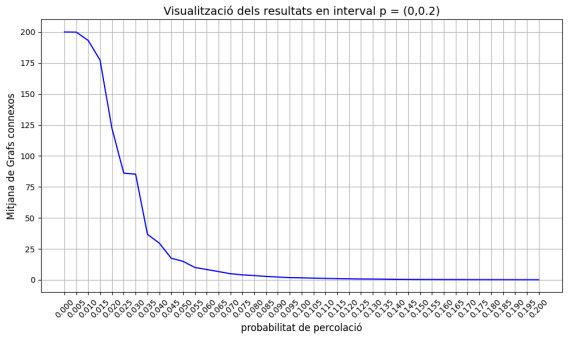
$$\frac{\#\ percolacions\ amb\ 1\ component\ connexa}{\#percolacions\ amb\ més\ components\ connexes}$$

   per a guardar el tant per 1 dels grafs que han percolat i continuent sent connexos. Seguidament, ho guardem en un fitxer .csv
4. Aquest experiment el repetim 15 cops, per a obtenir uns resultats més fiables.
5. Un cop obtenim els 15 fitxers amb les estadístiques en fem la mitjana per a cada valor de p i la mostrem en un gràfic.
6. Repetim els mateixos passos, però aquest cop per a la percolació d'arestes.

Resultats i Conclusions

Percolació de nodes	Percolació d'arestes
	

Observem que entre els valors 0.00 i 0.20 de p (aproximadament) es troba la transició de fase per ambdós casos, fem doncs una gràfica per a observar-ho millor limitant l'eix de les X a 0.2.

Percolació de nodes	Percolació d'arestes
	

Podem observar que per als dos casos la transició de fase es troba en valors molt baixos de p, però, tot i això, la percolació de nodes pren valors lleugerament més alts. A més podem veure que la percolació d'arestes té un pendent més elevat que la de nodes, la qual és més inestable. Això es pot donar pel fet que en l'eliminar vèrtex, és més difícil perdre la connectivitat d'un graf, ja que només pot passar en dos casos:

- Que s'esborrin els nodes adjacents a un node (en el millor cas són 2 nodes (cantonada), la majoria són 4 nodes adjacents i els costats són 3).
- Que un o més nivells quedin aïllats, és a dir, que una seqüència de nodes adjacents eliminats que va d'un costat de la graella a un altre.

Respecte a la transició de fase, per a la percolació de nodes hi seria a l'interval (0.01,0.125) i per a la percolació d'arestes es trobaria a l'interval (0.005,0.075), per tant, les probabilitats crítiques serien 0.0675 i 0.04 per a cada tipus de percolació respectivament.

Experiment 2:

El funcionament de l'experiment és el següent:

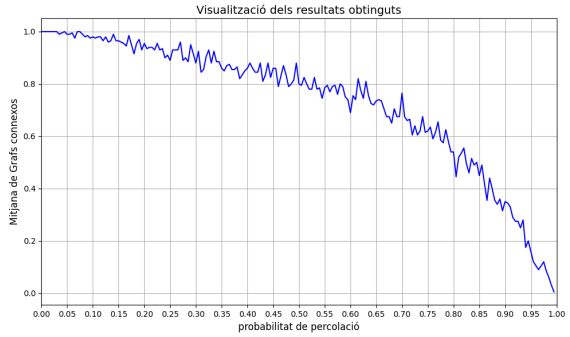
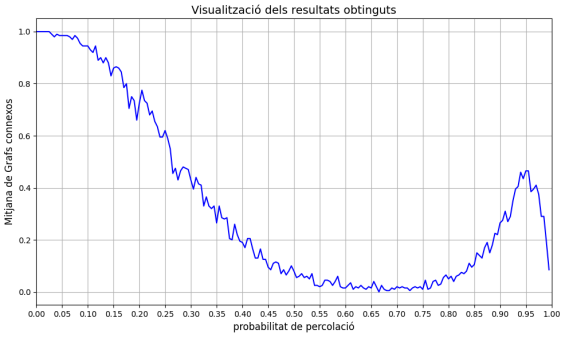
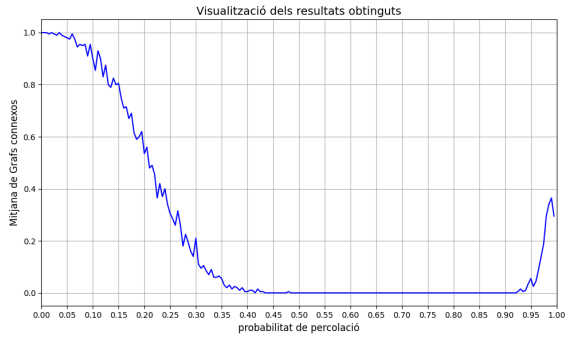
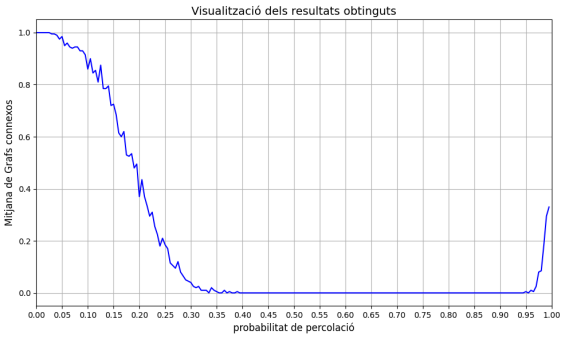
1. Generem un únic graf de mida n, començant amb mida 2*2
2. Apliquem percolació de nodes al graf per cada probabilitat dins del rang mencionat anteriorment i observem per a cada percolació si el graf només conté una component connexa o en té més d'una i guardem els resultats en un .csv.
3. Repetim el pas anterior 199 cops més, amb un total de 200 cops.
4. Guardem en un fitxer .csv la mitja dels 200 experiments per cada probabilitat p per a guardar el tant per 1 dels grafs que han percolat i continuen sent connexos. Seguidament, ho guardem en un fitxer .csv.
5. Del fitxer .csv obtenim un gràfic igual que els de l'experiment 1.

6. Repetim els passos 1 al 5 augmentant en 1 la mida (3*3, 4*4, 5*5, etc.) fins a 15*15
7. Repetim els mateixos passos, però aquest cop per a la percolació d'arestes.

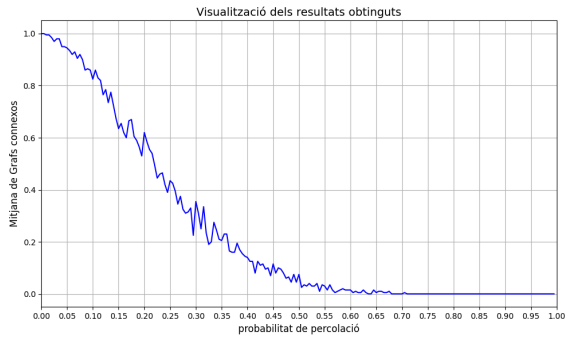
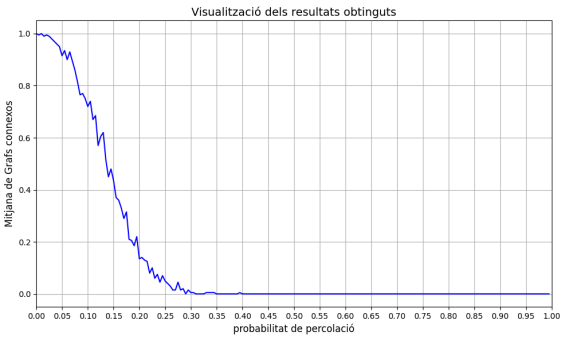
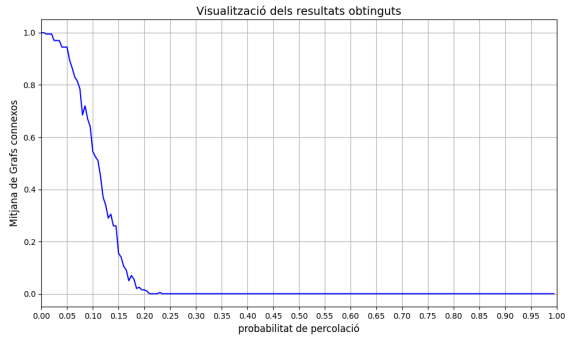
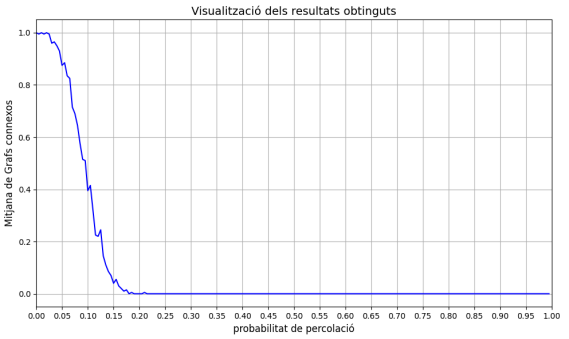
Resultats i Conclusions

Tot i haver dut a terme l'experiment per a cada mida, només veurem uns quants gràfics per a la comparació, ja que no volem ocupar massa espai amb les imatges. Per aquest motiu a l'annex es trobaran tots els gràfics.

Percolació de nodes:

n = 2*2	n = 5*5
	
n = 10*10	**n = 15*15**
	

Percolació d’arestes:

n = 2*2	n = 5*5
	
n = 10*10	**n = 15*15**
	

Amb aquests resultats fàcilment podem observar diverses propietats:

- Es confirma que els resultats amb la percolació són més inestables. A més podem observar per ambdós casos que com menor és la mida del graf més inestable és la gràfica, fins al punt que en la percolació de nodes de mida 2*2 no es podria determinar la transició de fase.
- Com major és la mida de la graella quadrada, abans es dona la transició de fase.
- A l’haver realitzat menys experiments per cada graf comparat amb l'experiment 1 (1*200 contra 200*15), la variabilitat dels resultats és major, encara que es pot seguir observant amb facilitat la transició de fase.
- A la percolació de nodes, amb probabilitats molt altes veiem un augment de connectivitat. Això ve donat que si només es queda un vèrtex en el graf seria connex.

4. Experimentació amb grafs connexos aleatoris

Disseny de l'experimentació

En aquest experiment es demana que a partir de la generació de grafs connexos aleatoris, fer un procés de percolació d'arestes i estudiar la transició de fase amb relació a les components connexes.

Entenem que un graf, inicialment connex, ha percolat si aquest després de passar per un procés de percolació ha perdut la seva propietat connexa, és a dir, no és connex. Per tant, per estudiar aquesta propietat amb una percolació d'arestes, hem plantejat la següent implementació.

1. **Un generador aleatori de grafs connexos**

Fent ús d'un generador de grafs aleatoris (Geometric_Generator.py), hem anat implementant diferents valors de "radi", fins que hem arribat a un punt on aquest generador ens donava consistentment grafs connexos. Aquests grafs es guarden en un fitxer .csv per la posterior lectura a partir de la funció de lectura de la nostra classe Graf.

2. **Comprovació que el graf generat és connex**

Fent ús de l'algorisme per determinar si el graf és connex, a partir d'una cerca de profunditat.

3. **La funció de percolació per arestes inicial**

Aquest procés es fa 100 vegades per cada graf, per poder calcular més correctament la transició de fase.

4. **Recopilació de la probabilitat de transició respecte a la probabilitat de percolació**

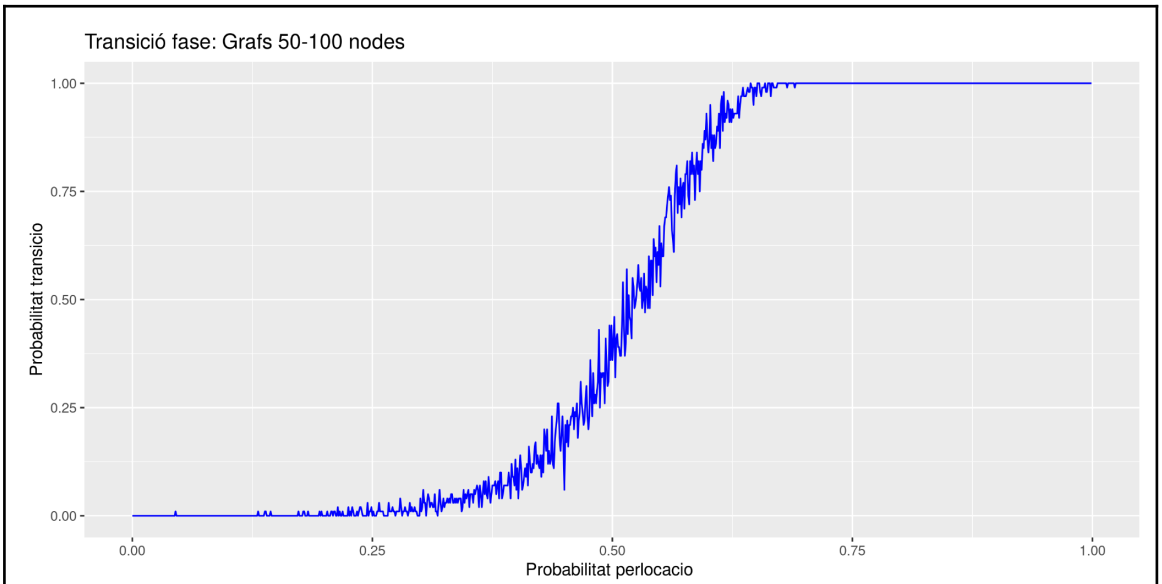
Les dades s'emmagatzemen en un fitxer .csv, per a facilitar el consegüent anàlisi de les dades trobades. La probabilitat de transició es calcula de la següent manera:

$$\frac{\#Grafs\ que\ han\ perlocat}{\#Total\ de\ grafs\ connexos(<=100)}$$

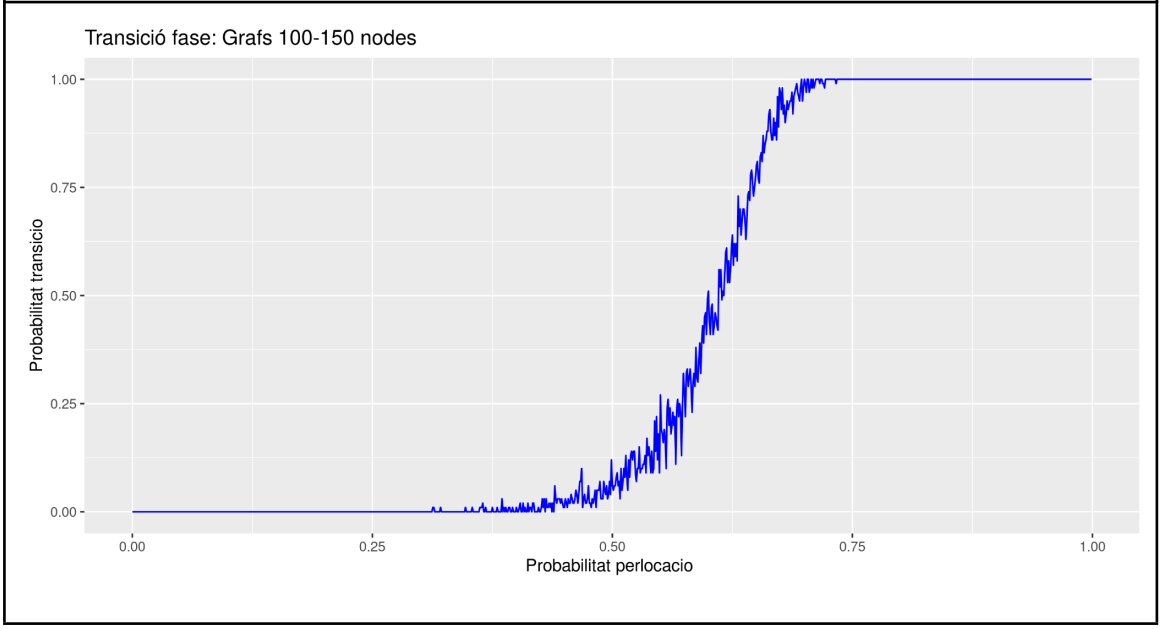
Els tres punts anteriors els hem implementat independentment tres cops, cada cop amb una mida, o sigui un nombre de nodes, dintre d'un interval diferent. Primer amb grafs de 50 a 100 nodes, després amb grafs de 100 a 150 i finalment amb grafs de 150 a 200. Per cada interval de nodes, hem fet 1000 iteracions que anaven augmentant la probabilitat de percolació, inicialment amb valor 0.001. Un cop recollit les dades, podem estudiar la transició de fase i com aquesta evoluciona a la vegada que augmenten el nombre de nodes que té cada graf.

Resultats i Conclusions

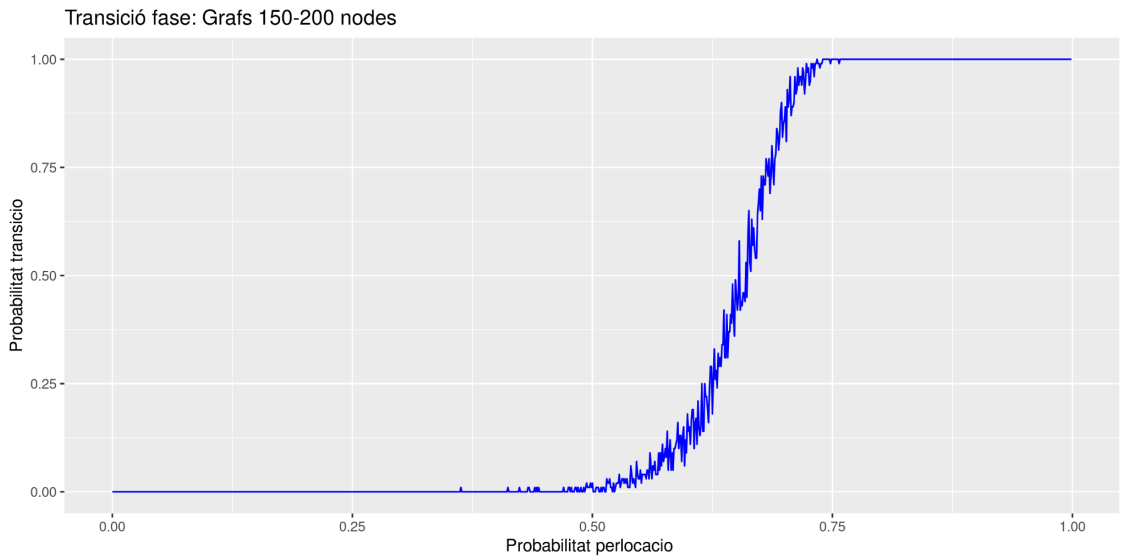
Els resultats de l'experimentació es poden veure en els següents gràfics:



Per aquest interval de nodes, s'observa una transició de fase a l'interval [0.25, 0.65]



Per aquest interval de nodes, s'observa una transició de fase a l'interval [0.375, 0.7]



Per aquest interval de nodes, s'observa una transició de fase a l'interval [0.5, 0.75]

Veiem que un cop creix el nombre de nodes, el valor de la probabilitat de percolació pel qual tenim un valor de transició de fase d'1.0, queda ben determinat al 0.75, es pot observar per grafs des de 100 a 200 nodes.

En canvi, on sí que veiem un canvi més gradual, és pel valor on s'inicia aquesta transició de fase. Observem que mentre augmenta el nombre de nodes del graf la probabilitat de percolació per la qual el resultat té una propietat màxima de ser un graf percolat. És a dir l'interval de la transició es va ajustant més i més, i es fa encara més precís quan augmentem el nombre de nodes.

Concloure doncs que l'interval de transició de fase serà de [0.5, 0.75], corresponent del trobat amb grafs de 150 a 200 nodes, encara que si augmentéssim el nombre de nodes, és possible que puguem acotar més. El que sí que podem afirmar és que la probabilitat crítica serà d'aproximadament de 0.625.

Problemes amb l'implementació.

Com hem vist anteriorment, si continuéssim fent més experiments amb grafs de mida superior podríem trobar un millor interval de transició de nodes.

El problema sorgeix en el temps que triga a fer-se un experiment complet per cada interval de nodes. Exactament, el temps que triga a generar aquests nodes aleatoris, tot i que tots els algoritmes tenen una complexitat en el temps basada en el nombre de nodes (o arestes).

Però el problema més greu el dona l'algoritme fet en Python, donat que per cada iteració del bucle intern fem escriptures de grafs en un .csv i els llegim en C++, on tenim un coll d'ampolla en la lectura i l'escriptura dels nostres grafs.

5. Experimentació amb graelles triangulars

Justificació

En aquest apartat se'ns demana que, dissenyem un experiment per a estudiar la probabilitat de transició de fase per a un tipus de grafs geomètric aleatori.
Hem triat les graelles triangulars com a graf geomètric perquè és un graf amb el qual era més senzill extrapolar l'experiment fet anteriorment amb grafs graelles quadrades NxN, ja que vam provar inicialment amb grafs tridimensionals i no va acabar de sortir bé.

Disseny d'Experimentació

Per aquest experiment hem hagut d'implementar:

1. Un Generador de Grafs graella triangulars
2. Retocar l'arxiu per a fer les percolacions per a que funcionés amb els grafs triats
3. Un compactador d'estadístiques
4. Un visualitzador de fitxers CSV (on teníem les estadístiques)

També hem fet un visualitzador de grafs per a poder entendre si el programa està percolant i comprovant bé els grafs.

El funcionament de l'experiment és el següent:

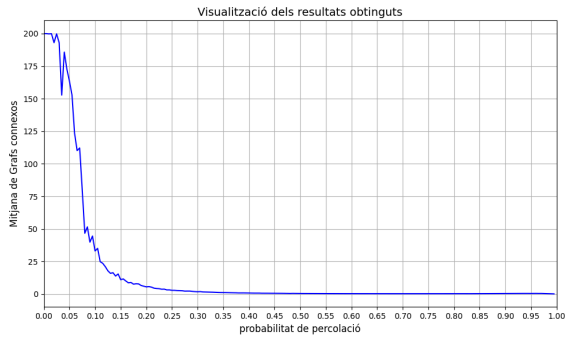
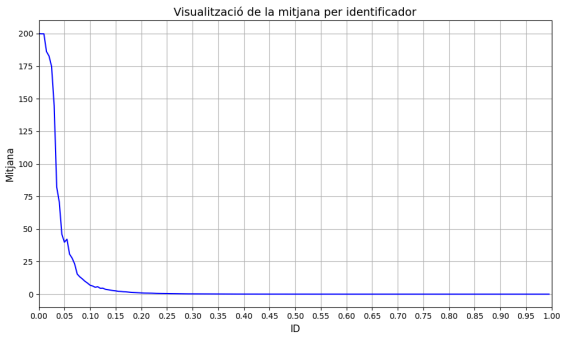
1. Generem un conjunt aleatori de 200 grafs graella triangulars d'alçada entre 2 i 10
2. Apliquem les percolacions de nodes i d'arestes a cada un dels grafs per cada probabilitat dins del rang [0,0.005,0.01,0.015,0.02, … , 0.985, 0.990, 0.995, 1.00] i veiem per a cada percolació si el graf només conté una component connexa o en té més d'una.
3. Per a cada probabilitat, operem aquesta fórmula:

$$\frac{\#\ percolacions\ amb\ 1\ component\ connexa}{\#percolacions\ amb\ més\ components\ connexes}$$

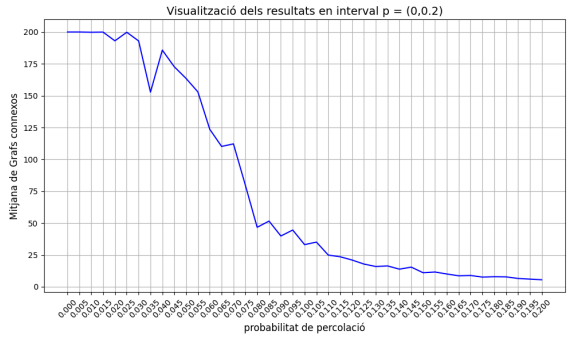
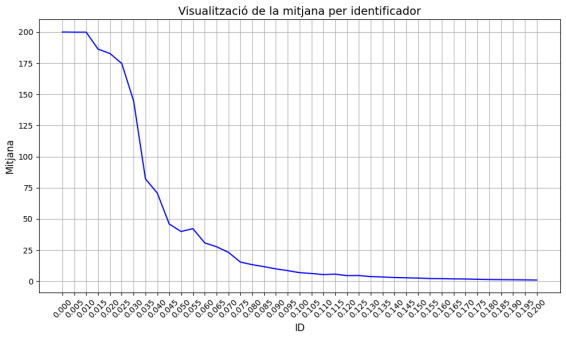
   per a guardar el tant per 1 dels grafs que han percolat i segueixen sent connexos. Seguidament ho guardem en un fitxer .csv.
4. Aquest experiment el repetim 15 cops, ja que hem vist que apareix un pic pels valors en el rang (0.040,0.065) i volíem veure si era per un cas de percolació concret o no.
5. Un cop obtenim els 15 fitxers amb les estadístiques en fem la mitjana per a cada valor de p i la mostrem en un gràfic.

Resultats i Conclusions

Veiem doncs els resultats:

Percolació de nodes	Percolació d'arestes
	

Observem que entre les dècimes 0.00 i 0.15 està la transició de fase, fem doncs una gràfica per a observar-ho millor limitant l'eix de les X a 0.2.

Percolació de nodes	Percolació d'arestes
	

Veiem que a la percolació de Nodes els valors són molt més inestables, això es deu a la topologia dels grafs graella triangulars. Durant la percolació de nodes podem tenir 2 casos en els quals ens quedin nodes aïllats:

1. Que els nodes adjacents desapareguin (en el millor cas són 2 nodes (cantonades), la majoria són 6 nodes adjacents i les bandes són 4).
2. Que el nivell en el qual estan s'aïlli, contant descendentment el cost és l'alçada del nivell inferior. (Millor cas 2 nodes)

Per això la percolació de nodes és més inestable, ja que la desaparició d'un sol node provoca que desapareguin moltes arestes al graf en un sol moviment i quan està en la fase de transició que desaparegui un node o no pot ser crític.

Finalment, ja podem definir realment quins són els valors de la transició de fase i quina és la probabilitat crítica, definim doncs que l'interval (0.015,0.075) pertany a la probabilitat de transició de fase i la probabilitat crítica és 0.034.

Procés d'Experimentació i Altres metodologies no executades

- **Model d'Ising i Grafs Tridimensionals**

Inicialment vam estar investigant i vam pensar d'estudiar la possibilitat de transició de fase per al Model d'Ising, que és un model que mesura l'entropia i el grau d'ordenació del graf. Vam estar provant i fins i tot vam arribar a implementar les funcions, però no vam acabar d'entendre els paràmetres amb els quals estàvem treballant i vam decidir fer-ho per nombre de components connexes.

De la mateixa manera, inicialment havíem pensat de fer l'experiment amb grafs tridimensionals, però havíem de canviar tot el model de grafs perquè ens ho pogués acceptar i vam decidir rebaixar-ho a dues dimensions i si ens sobrava temps intentar-ho.

- **Pic de valors entre 0.03 i 0.06**

Mentre executàvem l'experiment, vam observar pics de valors entre 0.03 i 0.06, en general molta inestabilitat a les primeres fases de transició. Fent recerca, vam arribar a la conclusió que es dona a les propietats dels grafs triangulars, on les cantonades són els punts més crítics del graf i que, com més nodes tenia, més estable era, ja que aquestes representaven un menor percentatge de nodes. Aquest efecte s'accentua quan fem la percolació de nodes perquè provoca que sigui encara més inestable.

Per això finalment ens vam decantar per no estructurar aquest experiment segons la mida dels grafs i barrejar-les totes, i per poder pal·liar aquest efecte, hem incrementat el nombre d'experiments i de grafs generats.

6. Conclusions del projecte i procés d'aprenentatge

Vam iniciar aquest estudi amb la premissa d'experimentar amb grafs per poder determinar si existeix una transició de fase en el procés de percolació de diferents tipus grafs. Durant el desenvolupament d'aquest projecte hem pogut anar experimentant amb estructures de grafs i la percolació. Millorant el nostre coneixement sobre aquest procés de fallida, que inicialment sembla purament probabilístic, però un cop hem pogut analitzar les dades que hem recollit, hem vist com darrere existeix propietats o tendències de comportament dels grafs al partir un procés d'eliminació o fallida tant de vèrtexs com d'arestes.

Aquesta tendència o propietat l'hem entès com a transició de fase, el trànsit del graf d'un estat no percolat, a mesura que s'augmenta la probabilitat de fallida de vèrtex o arestes, a un altre estat on passa a mantenir-se com un graf percolat durant el seguit de l'experimentació.

Amb l'anàlisi dels nostres resultats hem arribat a diferents conclusions per cada tipus de graf estudiat. Com en el cas dels grafs connexos aleatoris, els quals són el que pateixen una transició de fase més estable, que va acotant més l'interval de transició a la vegada que augmenta de mida.
En canvi, per grafs de graella quadrada sí que hem trobat més inestabilitat, sobretot per grafs de mides petites, i comparant els dos tipus de percolació hem vist que els resultats de la percolació de nodes esdevenen més inestables a causa de la propietat de connectivitat dels grafs.

No només hem treballat amb grafs típics sinó com s'ha vist en aquest estudi hem explorat els grafs triangulars descobrint la forma de poder implementar una generació d'aquest, entenen les seves característiques. I com reaccionen a un procés de percolació, on hem pogut veure la seva inestabilitat en certes mides a causa de les propietats de connectivitat dels grafs triangulars, i altres característiques peculiars de la transició de fase d'aquest tipus de grafs.

A més gràcies a la necessitat d'implementar aquesta pràctica, en aquest projecte hem pogut experimentar amb la construcció d'estructures de dades de representació de grafs, que ens en permet explorar les propietats i el funcionament de diferents tipus de grafs. A més explorar diferents formes de generació aleatòria d'aquestes estructures a partir de la generació aleatòria tant de vèrtex com arestes amb el repte de poder mantenir la seva connectivitat. I també la generació aleatòria en aquest cas de nombre reals aleatoris pel procés de percolació.

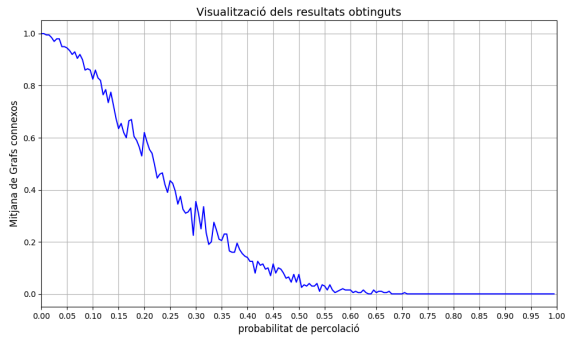
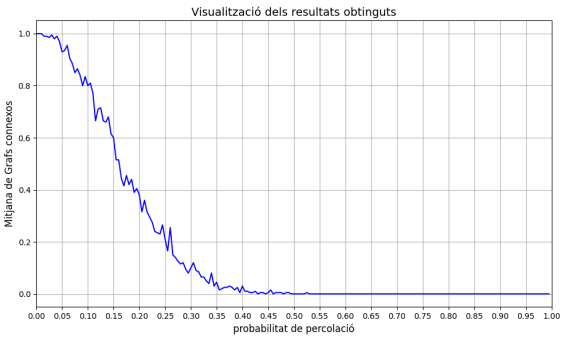
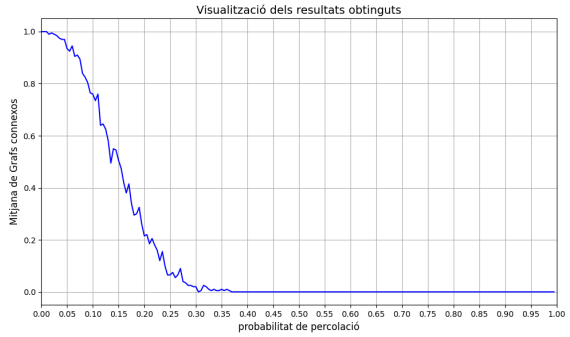
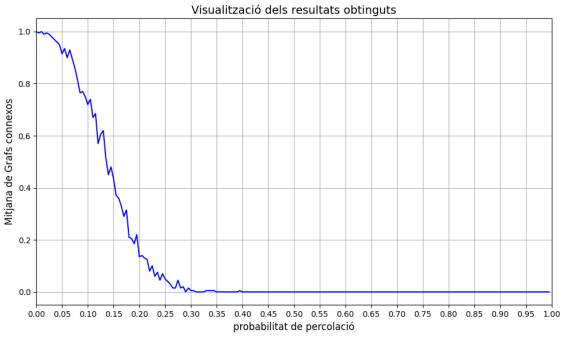
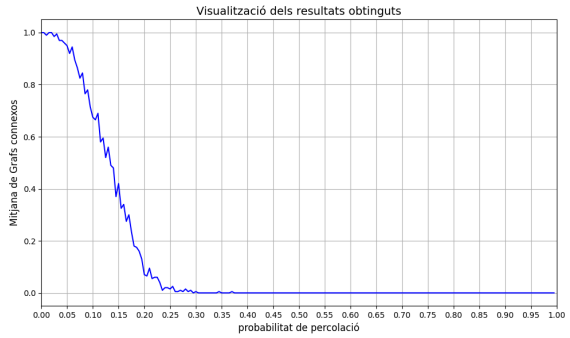
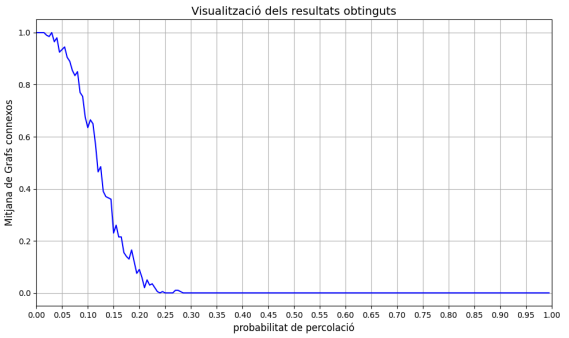
Finalment, com a proposta pròpia amb l'objectiu de poder entendre millor els grafs i assegurar el seu funcionament hem après a desenvolupar un visualitzador de grafs. Que esdevingut una eina molt útil sobretot a l'hora d'assegurar-nos del correcte funcionament tant de la nostra creació i lectura de grafs, com dels processos de percolació.

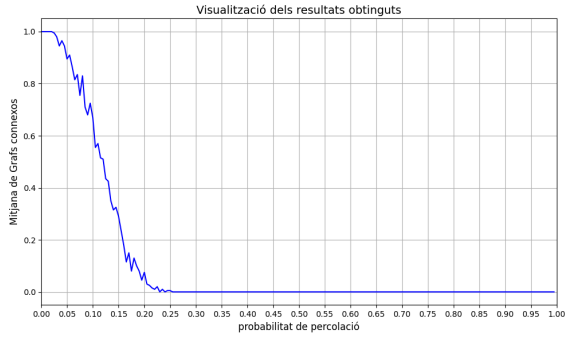
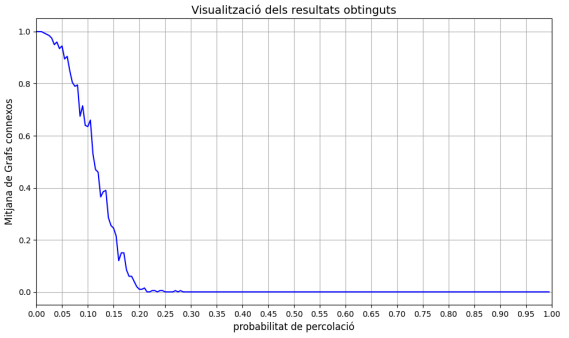
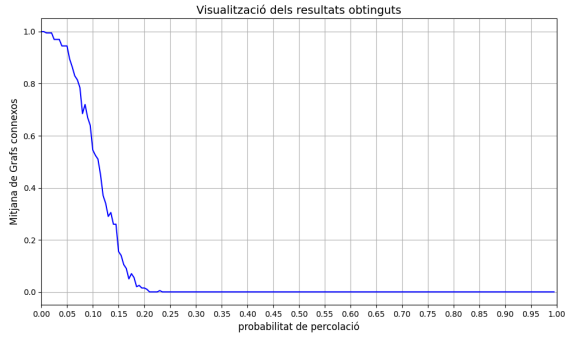
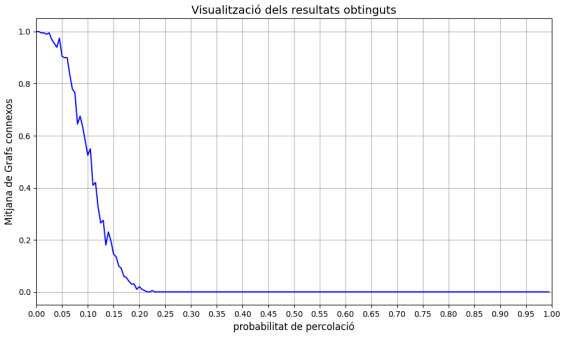
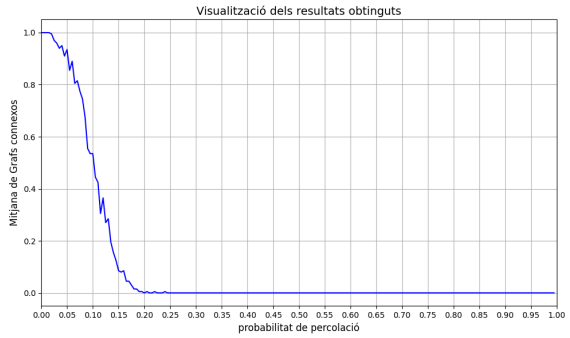
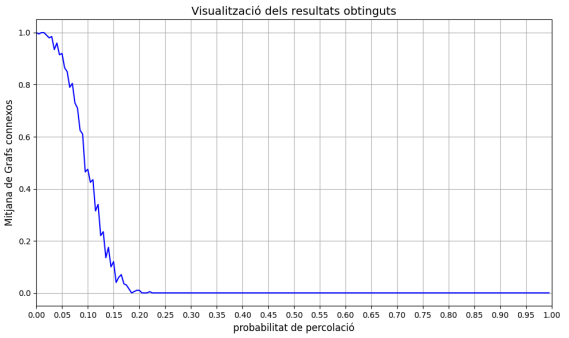
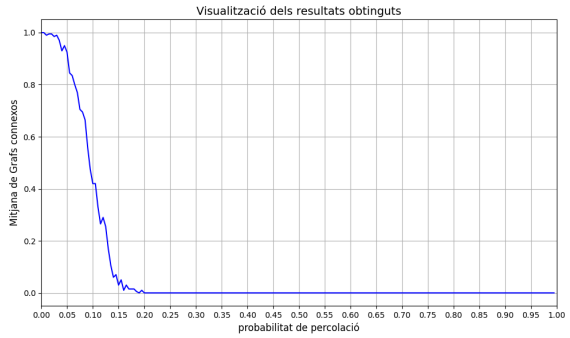
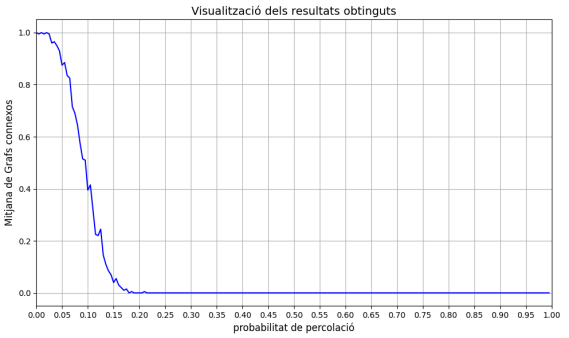
7. Annex

Experiments en grafs graella quadrats

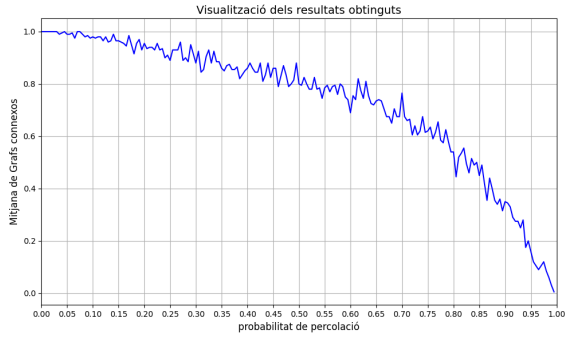
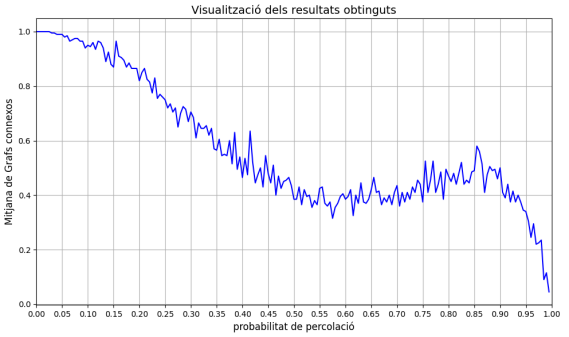
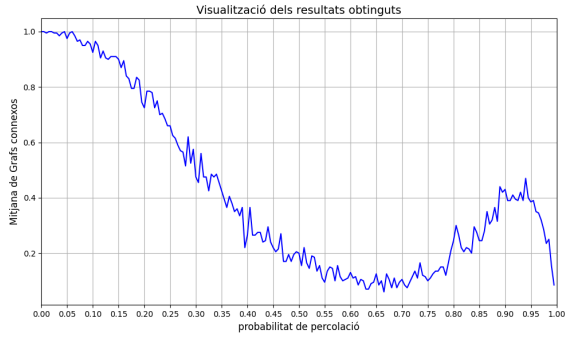
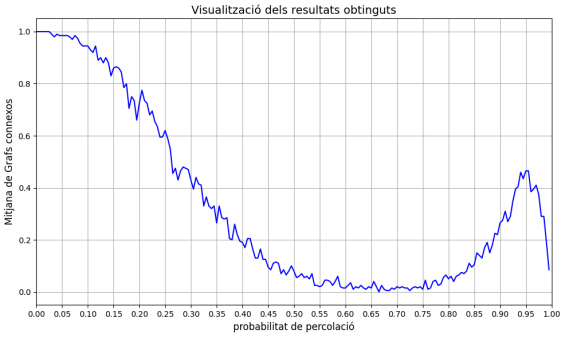
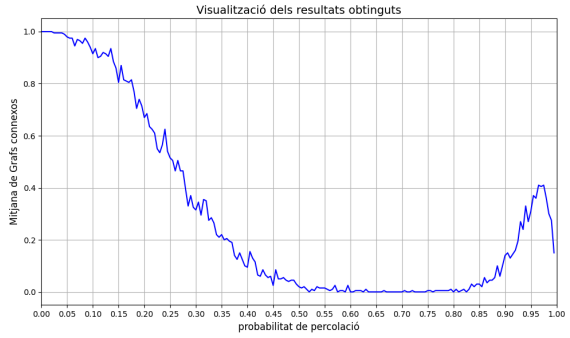
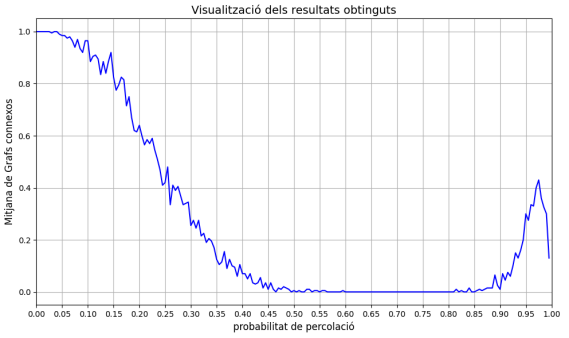
Gràfics experiment 2

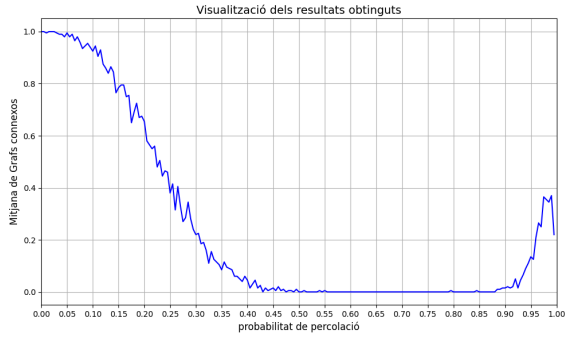
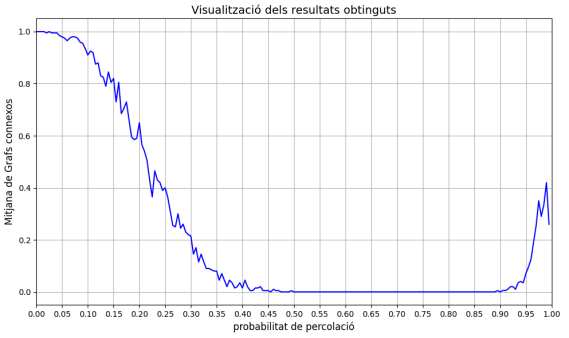
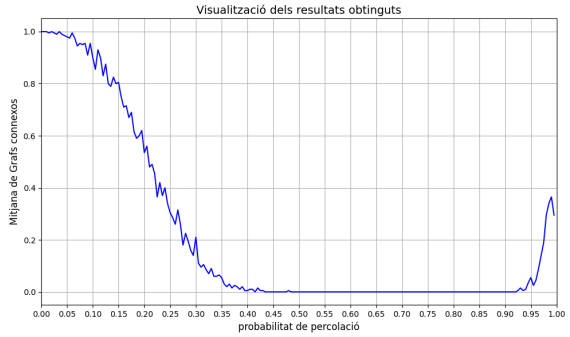
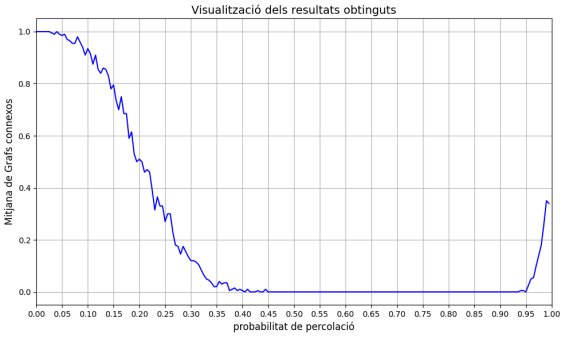
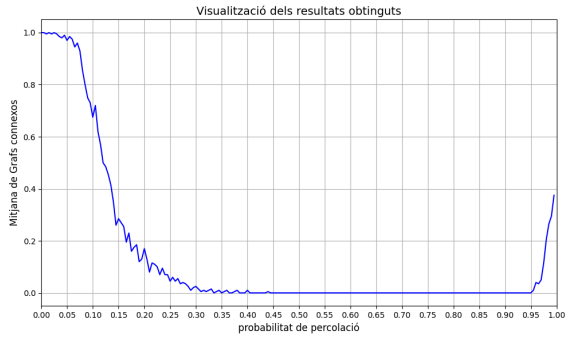
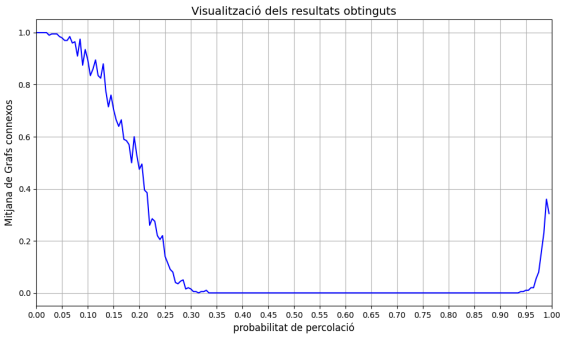
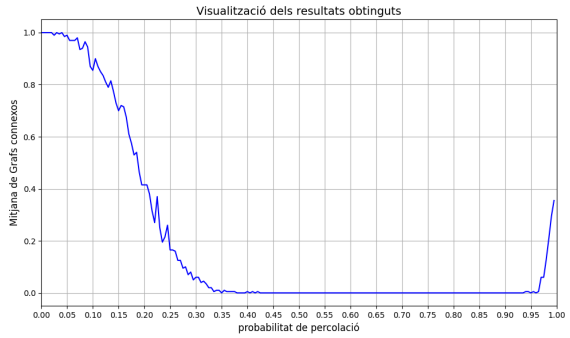
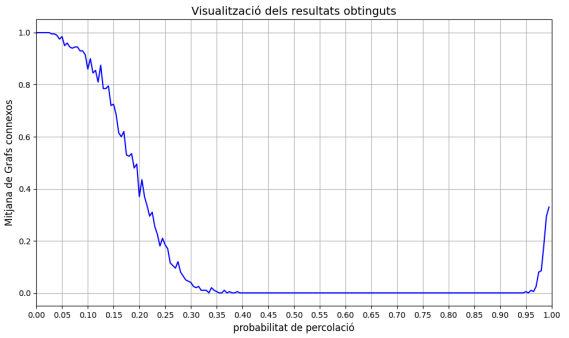
Percolació d'arestes:

n = 2	n = 3
	
n = 4	n = 5
	
n = 6	n = 7
	

n = 8	n = 9
	
n = 10	n = 11
	
n = 12	n = 13
	
n = 14	n = 15
	

Percolació de nodes:

n = 2	n = 3
	
n = 4	**n = 5**
	
n = 6	**n = 7**
	

n = 8	n = 9
	
n = 10	n = 11
	
n = 12	n = 13
	
n = 14	n = 15
	

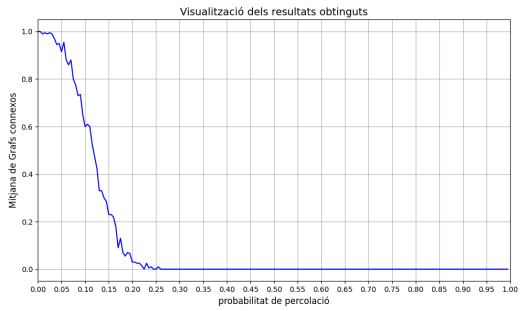
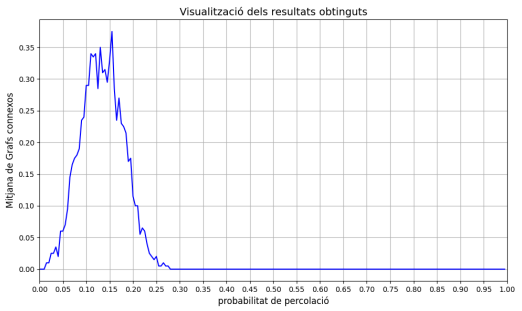
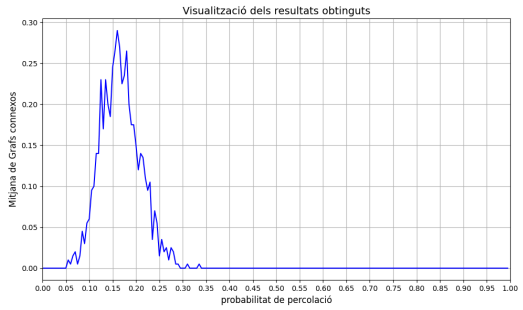
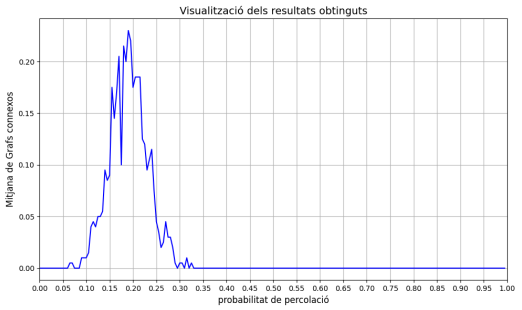
Experiment addicional

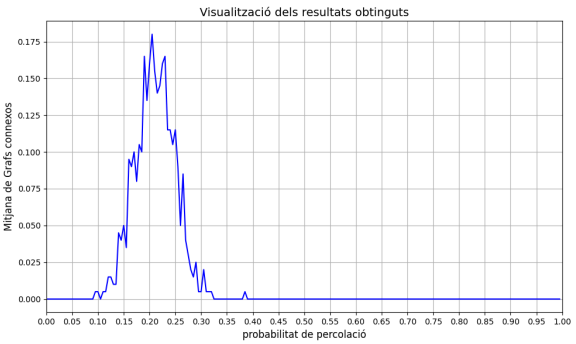
Per curiositat hem realitzat un petit experiment extra per a estudiar el comportament de la transició de fase de les percolacions, tant d'arestes com de nodes, segons el nombre de components connexes del graf. Per fer-ho hem seguit un procediment similar al del primer experiment, començant amb la percolació d'arestes:

- Fixem la mida del graf a 8*8 i realitzem 200 iteracions, fent la mitja i creant una gràfica exactament igual que al primer experiment.
- Ho repetim 4 cops més, però canviant lleugerament el codi perquè comptabilitzi quan té més components connexes. Per tant, acabarem tenint cinc gràfics, els quals mostren la transició de fase per cadascuna de les components connexes, des d'una fins a cinc.
- Realitzem els mateixos passos, però ara per a la percolació de nodes.

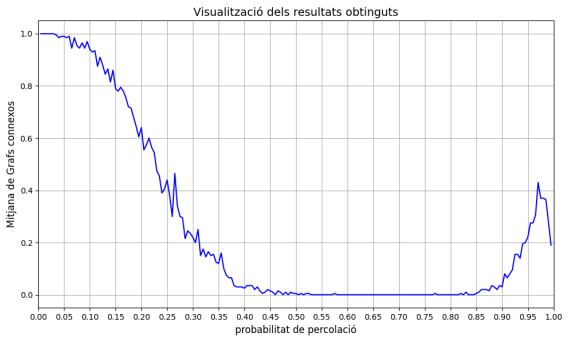
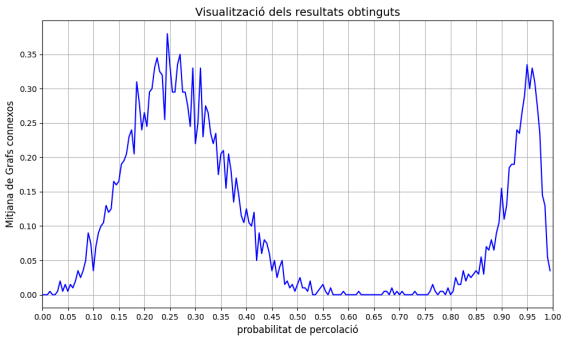
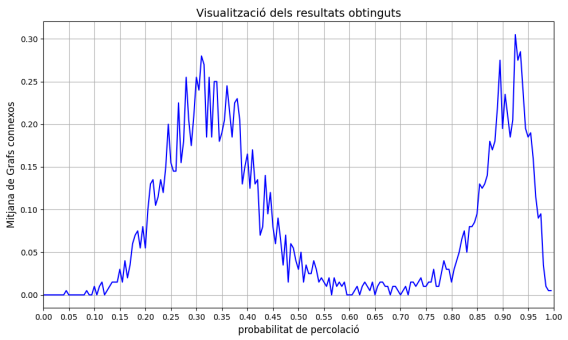
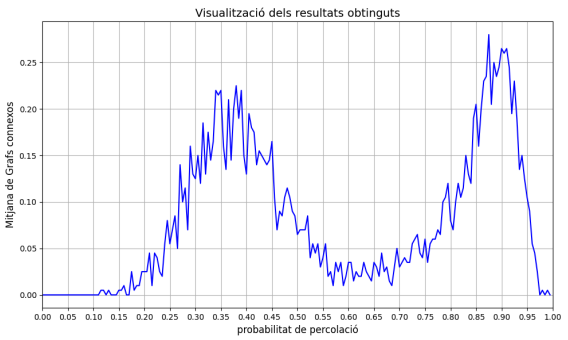
Resultats i conclusions:

Percolació d'arestes:

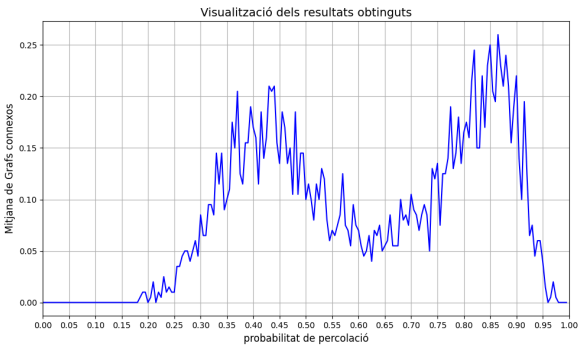
1 Component Connexa (CC)	2 CC
	
3 CC	**4 CC**
	

5 CC


Percolació de nodes:

CC = 1	CC = 2
	
CC = 3	**CC = 4**
	

CC = 5



Com podem observar, el comportament és molt diferent per la percolació d'arestes que per la de nodes, la qual addicionalment és molt més inestable com ja havíem vist als experiments anteriors.

A la percolació d'arestes sí que podem veure clarament per a quines probabilitats hi ha la transició de fase, és a dir per a quines probabilitats el graf resultant de la percolació té les components connexes determinades. La propietat que podem observar és que com més components connexes busquem, major és la probabilitat p necessària, tot i que l'augment de p per cada CC és molt petit, més del que esperàvem.

En canvi, a la percolació de nodes no només veiem un lleuger augment de p pels valors baixos, sinó que la propietat que havíem observat que es complia per valors alts s'accentua i va disminuint lleugerament per cada component. Per aquest motiu, com més nombres de CC busquem, més inestable es torna, fins al punt que no es pot determinar cap transició de fase, com podem veure en el cas per a cinc components connexes.

8. Bibliografia:

- *Jiatong Li, (2024, 26 marzo). "A brief overview of graph theory: Erdos-Renyi random graph model and small world phenomenon". Chicago University, Department of Mathematics, 2021. https://math.uchicago.edu/~may/REU2021/REUPapers/Li,Jiatong(Logen).pdf*

- *NetworkX — NetworkX documentation*. (s. f.). https://networkx.org/ - *Generació aleatòria dels diferents grafs emprats, en Python.*

- *CPlusPlus.com — Random Library documentation. (s. f.)* https://cplusplus.com/reference/random/ - *Generació aleatòria de nombre reals per l'algoritme percolació.*

- Sedgewick, R. S., & Wayne, K. W. (s. f.). Graphs. En *Algorithms* (4.ª ed.). https://algs4.cs.princeton.edu/40graphs/ - *Informació i ajuda de la implementació dels algoritmes.*

- *Forum - Grupo de C++ - Azul School. (s. f.). Azul School. https://www.azulschool.net/todos-los-grupos/grupo-de-c/forum/topic/generar-numeros-aleatorios-en-c/*

- *colaboradores de Wikipedia. (2023a, marzo 31). Teoría de la percolación. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_percolaci%C3%B3n*

- *std::system - cppreference.com. (s. f.). https://en.cppreference.com/w/cpp/utility/program/system*

- *Filesystem library (since C++17) - cppreference.com. (s. f.). https://en.cppreference.com/w/cpp/filesystem*

- *W3Schools.com. (s. f.). https://www.w3schools.com/cpp/ref_fstream_ofstream.asp*

- *GeeksforGeeks. (2024, 6 febrero). How to Read a File Using ifstream in C++? GeeksforGeeks. https://www.geeksforgeeks.org/read-file-using-ifstream-in-cpp/*

- *random_geometric_graph — NetworkX 3.4.2 documentation. (n.d.). Retrieved October 23, 2024, from https://networkx.org/documentation/stable/reference/generated/networkx.generators.geometric.random_geometric_graph.html*

- *igraph. (n.d.). Create a triangular grid graph · Issue #1842 · igraph/igraph. GitHub. Retrieved October 23, 2024, from https://github.com/igraph/igraph/issues/1842*

- *GeeksforGeeks. (2020, June 1). Draw a unstructured triangular grid as lines or markers in Python using Matplotlib. GeeksforGeeks. https://www.geeksforgeeks.org/draw-a-unstructured-triangular-grid-as-lines-or-markers-in-python-using-matplotlib/*